Fundamentos de IA y Machine Learning

Lección 5: Aprendizaje no supervisado - *Clustering*

eip ESCUELA INTERNACIONAL POSGRADOS

ÍNDICE

Aprendizaje no supervisado - *Clustering*

PRESENTACIÓN Y OBJETIVOS

Como se comentó previamente, el aprendizaje no supervisado es aquel empleado en bases de datos no etiquetadas. En esta lección, nos centraremos en los algoritmos de agrupamiento.

Objetivos

Al finalizar esta lección serás capaz de:

- Saber los fundamentos del aprendizaje no supervisado y, en concreto, del *clustering*.
- Conocer las características fundamentales del *clustering* particional, jerárquico y basado en densidades.
- Aplicar los distintos tipos de agrupamiento e interpretar los resultados.

1. Introducción y conceptos clave

1.1. Aprendizaje no supervisado

Los métodos de aprendizaje no supervisado no necesitan datos etiquetados, sino que tratan de descubrir la estructura de estos. Puede ser un objetivo por sí solo o un medio para un fin. Existen distintos objetivos enmarcados dentro del aprendizaje no supervisado:

- **Agrupación:** dados dos ejemplos sin etiquetar, agruparlos siguiendo algún criterio predefinido.
- **Generación de jerarquías:** dados unos datos en un mismo nivel, generar las jerarquías que organicen dichos datos.
- **Reducción de dimensionalidad:** dados unos datos, reducir la dimensión o número de atributos que caracterizan dichos datos.
- **Visualización:** dados unos datos con representación compleja, permitir su visualización.

	Features			
Objects	F1	F2	F3	...
obj1	.	.	.	...
obj2	.	.	.	...
⋮	⋮	⋮	⋮	⋱

Figura 1.1: Ejemplo de base de datos de aprendizaje no supervisado.

1.2. Agrupamiento

El objetivo del agrupamiento es agrupar las instancias o patrones creando *clusters* similares. Es decir, segmentar una población heterogénea en un número de subgrupos homogéneos o *clusters.* Se debe entender a qué *cluster* pertenecen los objetos y cuáles son las características comunes de cada *cluster*. Así, un *cluster* se define como un grupo o conjunto de objetos similares entre sí y distintos a lo de otro *cluster.*

Una característica común con respecto a la clasificación es que ambos tratan de asignar la clase o *cluster* apropiado a un determinado ejemplo. Sin embargo, en *clustering* no existen clases predefinidas, por lo que el objetivo es agrupar los datos, en lugar de desarrollar clasificadores (no se realizarán particiones de los datos y será más complicada la evaluación de una determinada metodología).

Por ejemplo, ¿cuál sería la forma natural de agrupar a los personajes de los *Simpsons*?

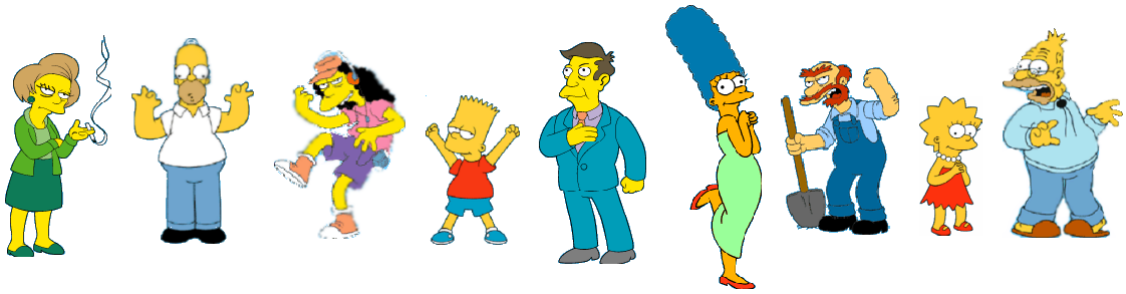

Figura 1.2: Personajes de los *Simpsons*.

Todo **depende de las características** en las que nos basemos para realizar la agrupación. Si utilizamos la variable “sexo”, tendremos:

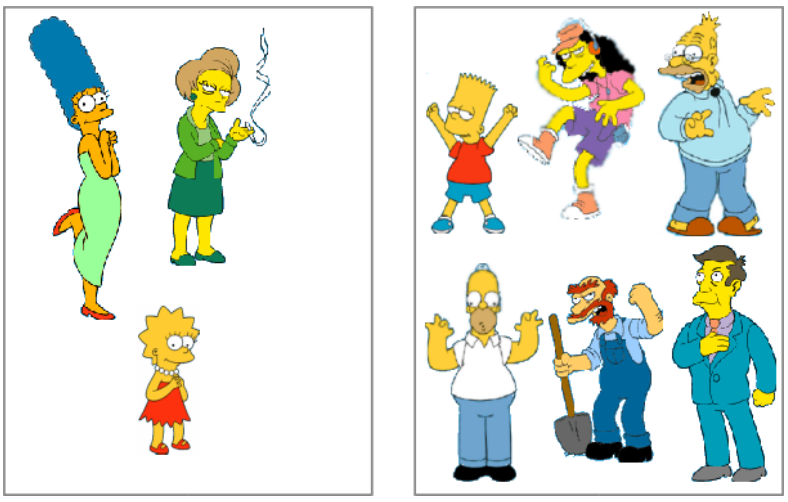

Figura 1.3: Personajes de los *Simpsons* agrupados por "sexo".

Sin embargo, si agrupamos teniendo en cuenta si pertenecen o no la familia *Simpsons*, tendríamos los siguientes dos grupos:

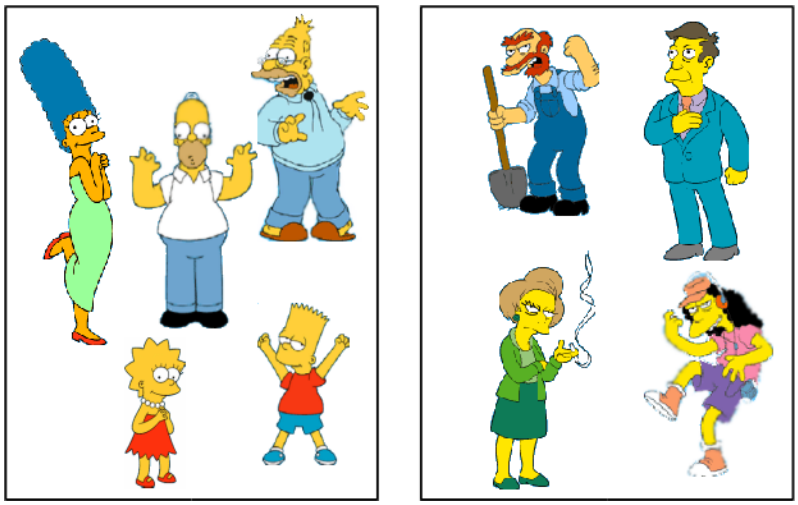

Figura 1.4: Personajes de los *Simpsons* agrupados por "familia".

De esta forma, se deduce que dependiendo del objetivo una opción será mejor que otra. El *clustering* es **subjetivo**.

1.3. Aplicaciones del agrupamiento

Existen distintos campos donde se aplica el agrupamiento:

- **Marketing:** identificar grupos de clientes con comportamiento similar (segmentación de mercado) o posicionamiento de productos.
- **Internet:** agrupamiento de textos, vídeos, imágenes...
- **Genética:** agrupamiento de genes para inferir estructuras en la población.
- **Redes sociales:** identificar comunidades.
- **Sistemas de recomendación:** recomendar productos similares a los gustos personales.
- **Ciencias sociales:** minería de datos educativos, tipologías de opiniones, etc.

1.4. Medidas de similitud

La propiedad más importante que debe verificar un *cluster* es que haya más cercanía entre la instancias que están dentro del *cluster* que respecto a las que están fuera del mismo.

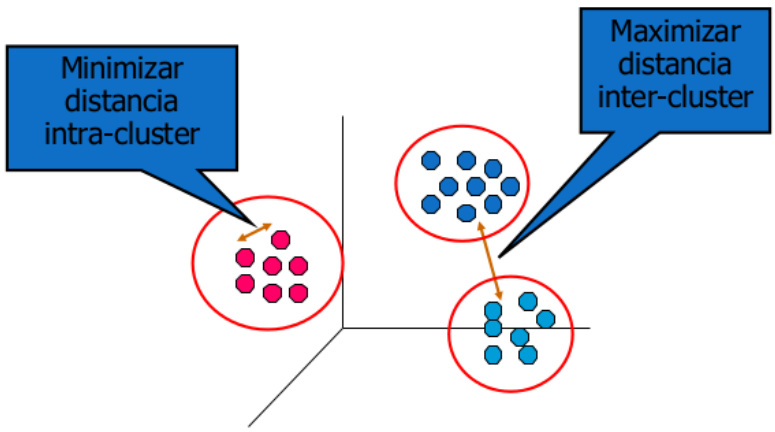

Figura 1.5: Minimizar la distancia *intra-cluster* y maximizar distancia *inter-cluster*.

Hablamos de similitud entre instancias, pero ¿qué es la similitud y cómo la medimos? Usualmente, se expresa en términos de distancias, dónde $d(i,j) > d(i,k)$ nos indica que el objeto i es más parecido a k que a j.

La definición de la métrica de similitud/distancia será distinta en función del tipo de dato y de la interpretación semántica que nosotros hagamos. En otras palabras, la similitud entre objetos es subjetiva.

En la lección 2, ya vimos distintas métricas de distancia que pueden ser utilizadas en el *clustering*:

- **Distancia euclídea:** raíz cuadrada de las diferencias al cuadrado de las características de dos patrones.

$$d_E(X_1, X_2) = \sqrt{\sum_{i=1}^{K} (X_{1i} - X_{2i})^2}$$

- **Distancia Manhattan:** suma de las diferencias absolutas de las características de dos patrones.

$$d_M(X_1, X_2) = \sum_{i=1}^{K} |X_{1i} - X_{2i}|$$

- **Distancia Ajedrez:** mayor de sus diferencias a lo largo de cualquiera de sus dimensiones características.

$$d_A(X_1, X_2) = max_{i=1}^{K}(|X_{1i} - X_{2i}|)$$

Los resultados del agrupamiento dependerán de la distancia elegida, del tipo de *clustering* aplicado y del número de *clusters* seleccionados.

1.5. Evaluación de los métodos de agrupamiento

La evaluación del agrupamiento depende de lo que estemos buscando. Por ejemplo, habrá situaciones en las que nos interese evitar descubrir *clusters* donde solo haya ruido, otras veces, por ejemplo, será comparar dos técnicas de agrupamiento.

Existen criterios externos, que consisten en aportar información adicional. No obstante, para evaluar el *clustering* se suelen utilizar criterios internos (a partir de los propios datos) para comparar métodos de *clustering* o para estimar el número de *clusters*. Una de las métricas más utilizadas es la suma del error cuadrático (*SSE*) que calcula la suma de las distancias al cuadrado de cada patrón con respecto a su centroide. El centroide sería el punto medio de todos los objetos pertenecientes a un *cluster*. El *SSE* se define como:

$$SSE = \sum_{i=1}^{n} (\boldsymbol{x_i} - \boldsymbol{\mu_{x_i}})^2$$

Siendo $\boldsymbol{x_i}$ un patrón y $\boldsymbol{\mu_{x_i}}$, el centroide de dicho patrón. El principal problema de esta métrica es que a mayor número de *clusters*, el *SSE* es más pequeño, y al tratarse de un problema de minimización tendríamos que saber cuándo no tiene sentido seguir creando *clusters* a pesar de disminuir el error. En otras palabras, determinar el número óptimo de *clusters* a partir del cual la disminución del error no merece la pena. Para ello, se utiliza la técnica del "codo" que consiste en representar el error frente al número de *clusters* y observar cual es el valor adecuado en el que se produce un "codo" en la gráfica. En el ejemplo siguiente, el codo sugiere que el número de *clusters* óptimo es dos.

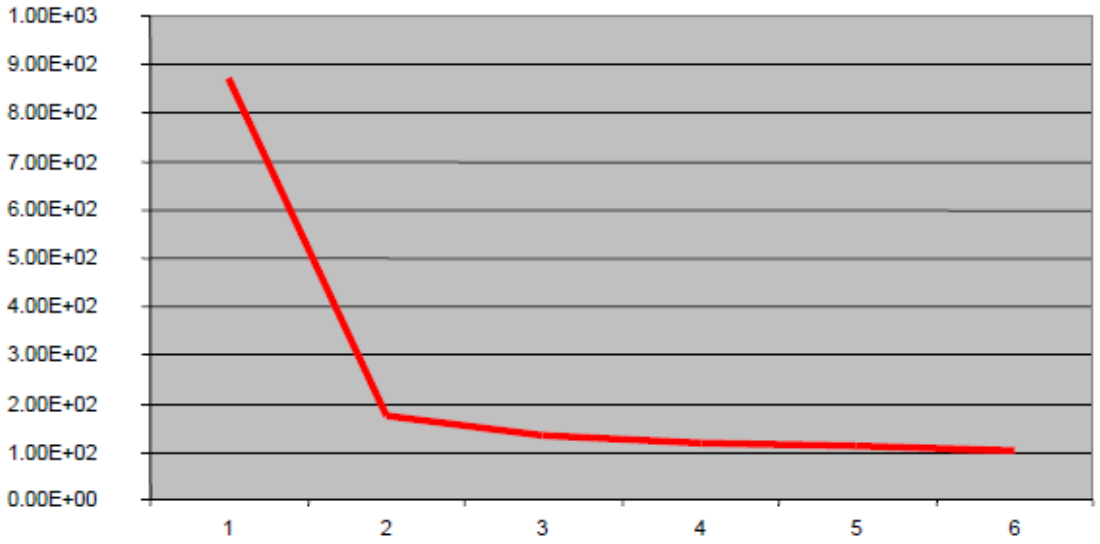

Figura 1.6: Técnica del "codo" para determinar el número de *clusters* óptimo.

Aun así, existen otras medidas que optimizan la cohesión de los grupos, la separación, coeficientes de silueta, ...

1.6. Requisitos para el algoritmo "perfecto"

Aunque es muy complicado obtener un algoritmo perfecto, se persigue encontrar aquél que:

- Sea escalable al tamaño de los datos.
- Maneje distintos tipos de datos.
- Identifique *clusters* con formas arbitrarias.
- Tenga un número mínimo de parámetros.
- Sea tolerante al ruido y a *outliers*.
- Sea independiente al orden de presentación de los patrones de entrenamiento.
- Permita trabajar con espacios con muchas dimensiones diferentes.
- Sea interpretable.

1.7. Tipos de *clustering*

Existen distintos tipos de agrupamiento:

- **Particionales:** construyen distintas particiones de acuerdo a un criterio.
- **Jerárquicos:** crean una descomposición jerárquica usando algún criterio.
- **Otros:** de densidad, basados en modelos, etc.

Veremos que distintos algoritmos llevarán a distintos *clusters*. Es muy importante estudiar bien el problema, los datos y los distintos algoritmos para decidir cuál es el más apropiado.

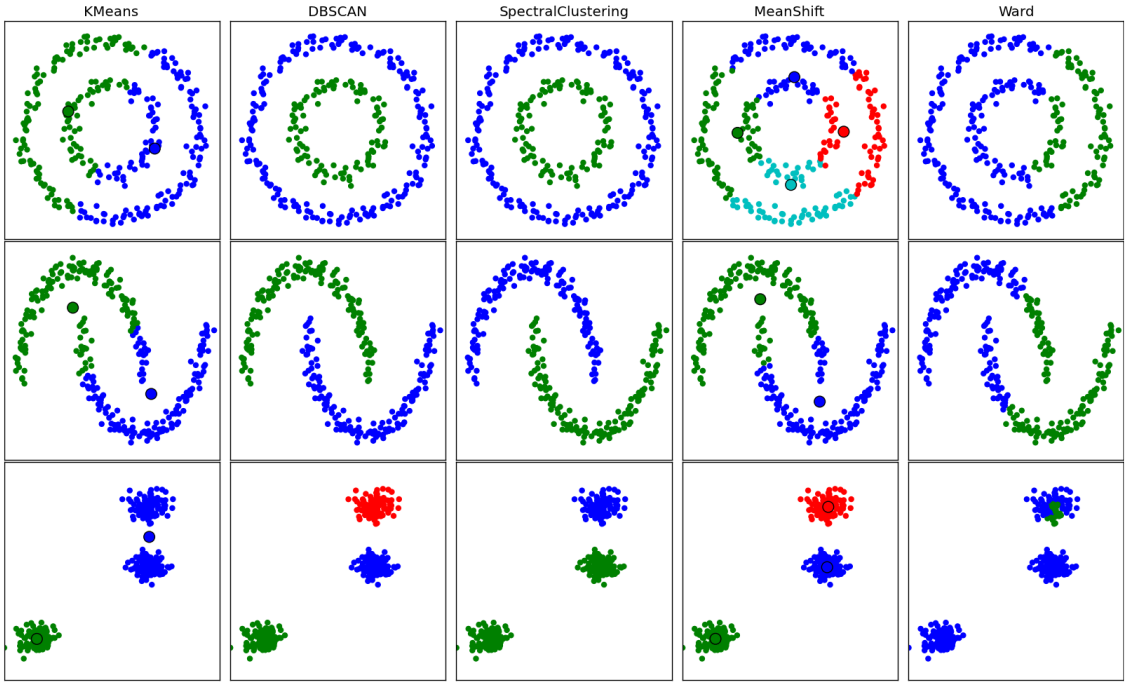

Figura 1.7: Resultados de aplicar distintos tipos de *clustering*.

2. AGRUPAMIENTO PARTICIONAL

Este tipo de *clustering* se basa en particionar un conjunto de N datos en k grupos de acuerdo a un criterio.

2.1. Algoritmo k-medias

En este algoritmo se asume que se conoce o se cree conocer el número de *clusters* k. Por tanto, dado un valor de k, la idea es encontrar la partición que optimice el criterio de particionamiento. Cada grupo tiene asociado un centroide y se busca que cada punto o patrón esté lo más cerca posible de su centroide, utilizando cualquier métrica de distancia. Iterativamente, se van actualizando estos centroides en función de las asignaciones de cada punto a los distintos grupos, proceso que se repite hasta que los centroides dejan de cambiar. Por tanto, el procedimiento general sería:

1. Seleccionar las k semillas iniciales, es decir, los centroides iniciales: se pueden utilizar patrones existentes o puntos dentro del espacio de los patrones.
2. Asignar cada patrón al *cluster* (centroide) más cercano: usando una de las distancias vistas anteriormente.
3. Actualizar los centroides: el algoritmo k-medias calcula el centroide como el punto medio de todos los patrones del *clusters*. Existen otras variantes como el algoritmo k-medianas que utiliza, como su nombre indica, la mediana en lugar de la media.
4. Repetir el proceso hasta que los centroides no cambien o la asignación de los patrones sea la misma, lo que indica que el algoritmo ha convergido.

2.2. Ventajas e inconvenientes

La principal ventaja de este algoritmo es que tiene una buena eficiencia. Sin embargo, posee una serie de desventajas que son recomendables tener en cuenta.

- Se necesita saber el número de agrupamientos k.
- Incapacidad de detectar ruido u *outliers*.
- El resultado final **depende de la inicialización**, y además es muy sensible a esta.

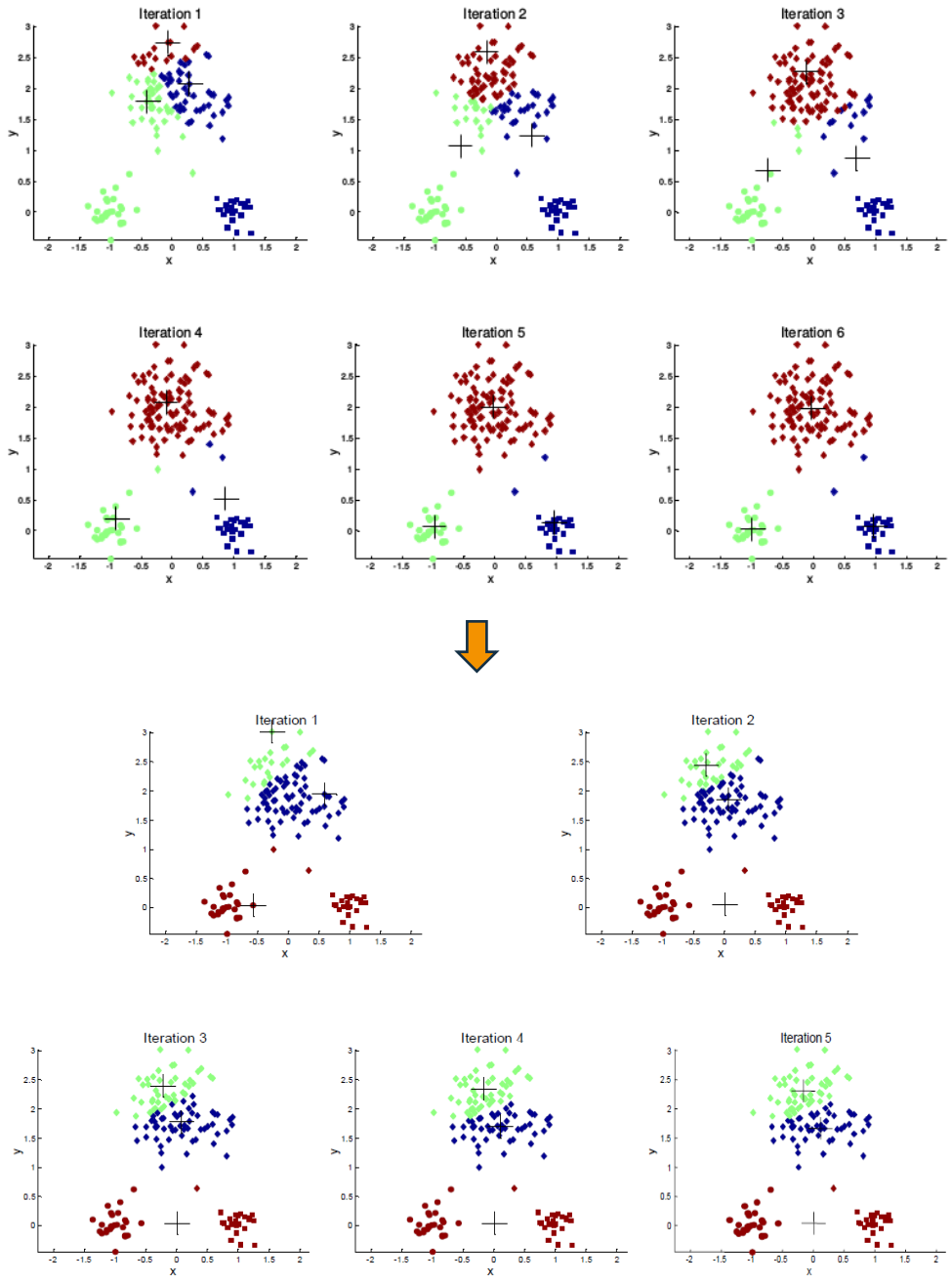

Figura 2.1: Resultados de aplicar k-medias con distintas semillas.

De este modo, con frecuencia finaliza en un **óptimo local.**

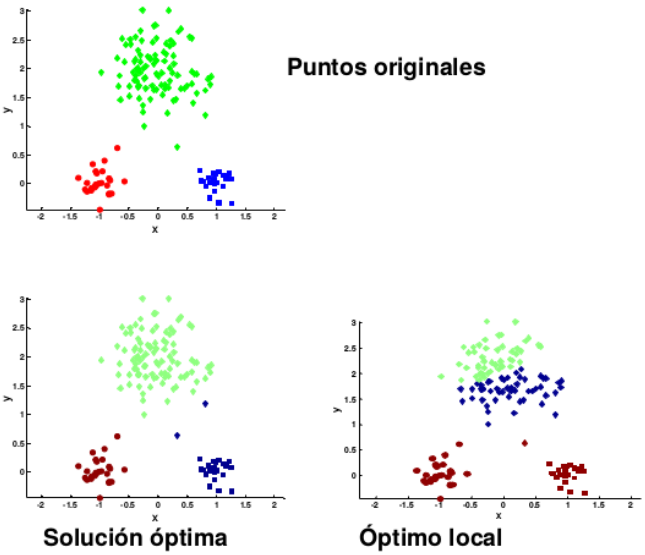

Figura 2.2: Óptimo local vs solución óptima global.

No recomendado para *clusters* de distinto tamaño.

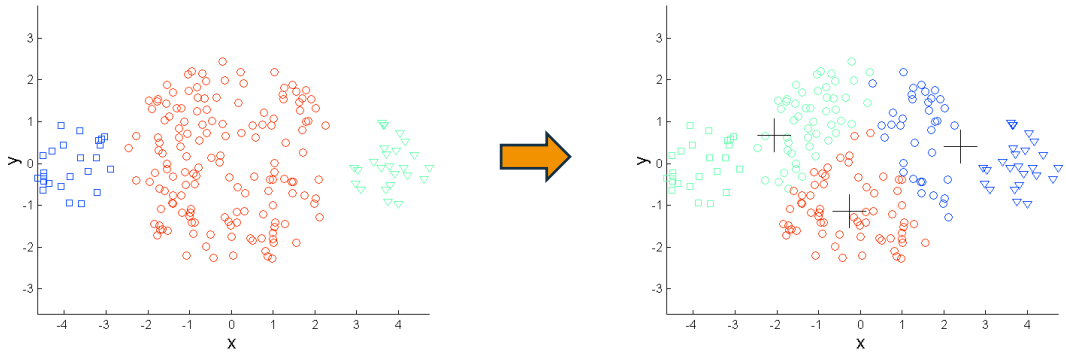

Figura 2.3: *Clusters* de distinto tamaño.

No recomendado para *clusters* de distinta densidad.

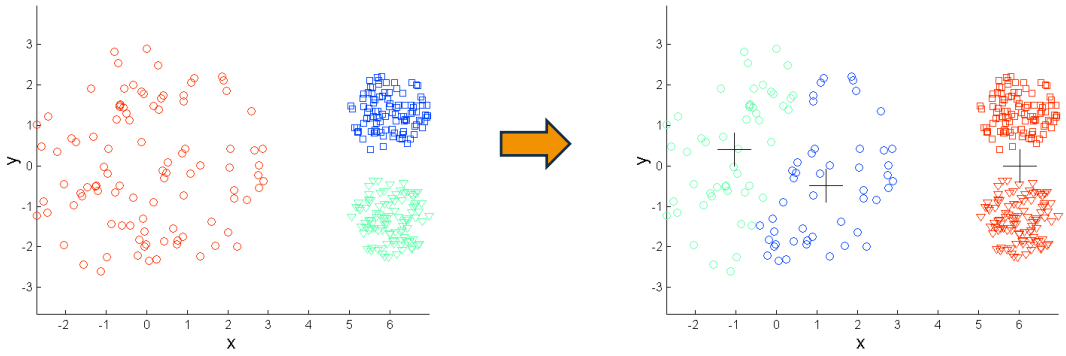

Figura 2.4: *Clusters* de distinta densidad.

Sólo recomendado para ***clusters* convexos**.

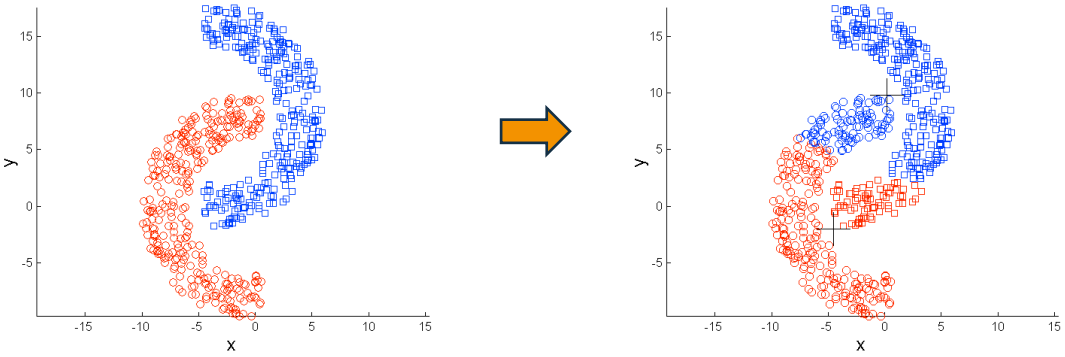

Figura 2.5: *Clusters* no convexos.

2.3. Ejemplo de aplicación

Se pide agrupar un total de 8 patrones bidimensionales en tres *clusters* ($k = 3$). Los patrones son los siguientes: A1 (2,10), A2(2,5), A3(8,4), A4(5,8), A5(7,5), A6(6,4), A7(1,2) y A8(4,9). Los centroides iniciales son los puntos A1, A4 y A7. La métrica de distancia utilizada será la distancia euclídea.

Se pide:

1. Representar los *clusters* creados y la posición de los centroides después de cada iteración.
2. El valor de la métrica *SSE*.

Solución:

Estado inicial

C1: (2,10) C2: (1,2) C3:(5,8)

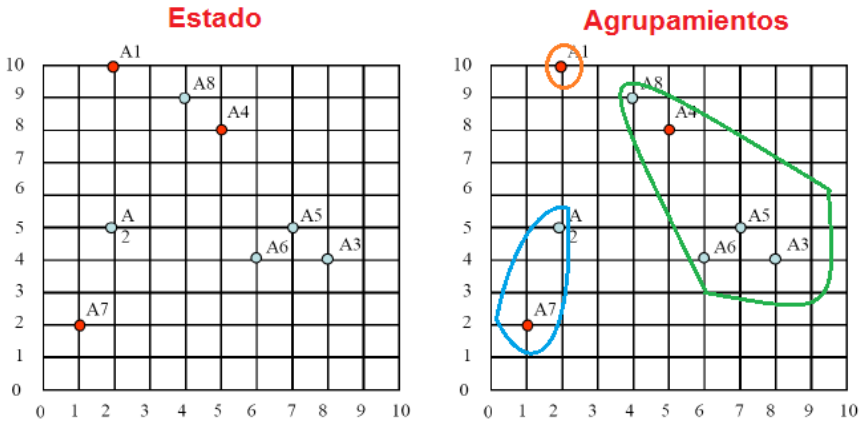

Figura 2.6: Estado inicial *k*-medias.

Iteración 1

Recalculamos centroides:

C1: (2,10)

C2: ((1+2) /2, (2+5) /2) = (1.5,3.5)

C3: ((4+5+6+7+8) /5, (9+8+4+5+4) /5) = (6,6)

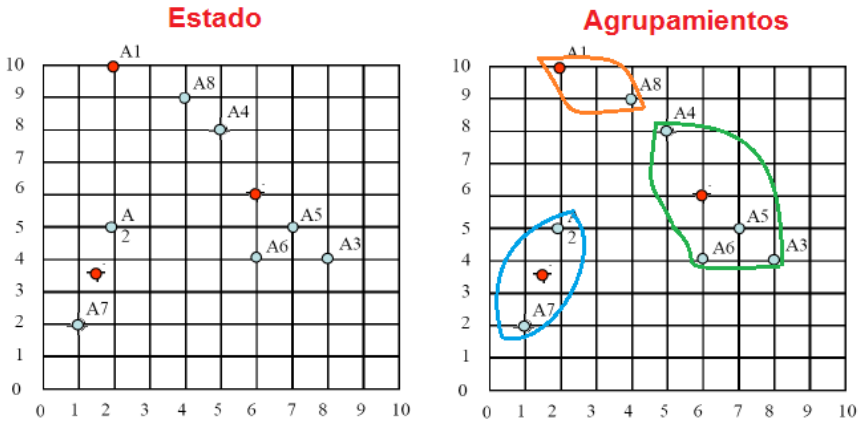

Figura 2.7: Iteración 1 *k*-medias.

Iteración 2

C1: ((2+4) /2, (10+9) /2) = (3,9.5)

C2: ((1+2) /2, (2+5) /2) = (1.5,3.5)

C3: ((5+6+7+8) /4, (8+4+5+4) /4) = (6.5,5.25)

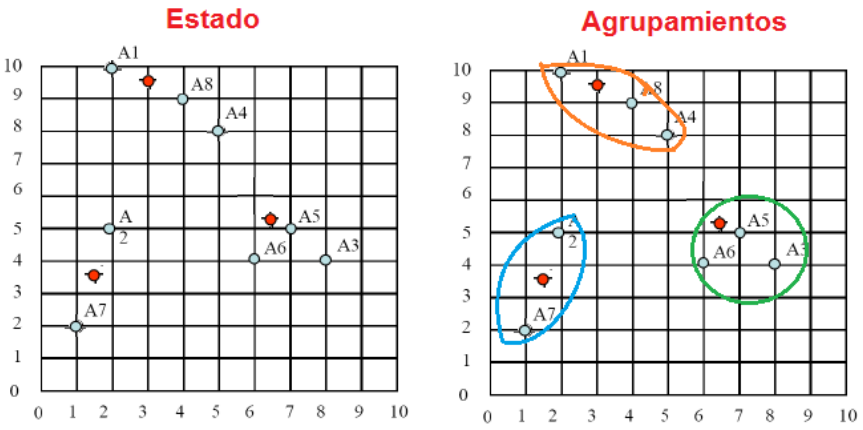

Figura 2.8: Iteración 2 *k*-medias.

Iteración 3

C1: ((2+4+5) /3, (10+9+8) /3) = (3.66,9)

C2: (1.5,3.5)

C3: ((6+7+8) /3, (4+5+4) /3) = (7,4.33)

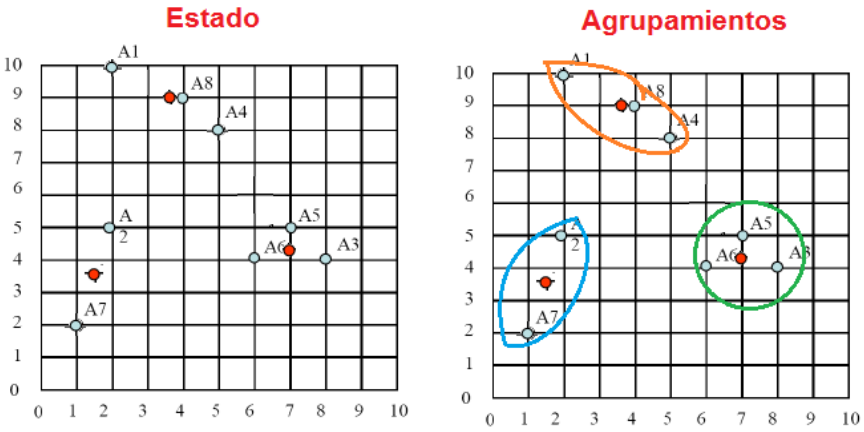

Figura 2.9: Iteración 3 *k*-medias.

El algoritmo pararía aquí, ya que los *clusters* creado son iguales a los de la iteración anterior. Para calcular el *SSE*, se resumen los cálculos en la siguiente tabla.

Punto	*Cluster*	Distancia al centroide
A1 (2,10)	*Cluster* 1: C1(3.66,9)	1.938
A2 (2,5)	*Cluster* 2: C2(1.5,3.5)	1.5811
A3 (8,4)	*Cluster* 3: C3(7,4.33)	1.053
A4 (5,8)	*Cluster* 1: C1(3.66,9)	1.672
A5 (7,5)	*Cluster* 3: C3(7,4.33)	0.67
A6 (6,4)	*Cluster* 3: C3(7,4.33)	1.053
A7 (1,2)	*Cluster* 2: C2(1.5,3.5)	1.5811
A8 (4,9)	*Cluster* 1: C1(3.66,9)	0.34
	SSE	**14.3333**

3. Agrupamiento jerárquico

Este tipo de *clustering* se basa en medir la distancia entre *clusters*.

3.1. Dendograma

El dendograma nos permite representar la similitud entre los objetos y el cómo se van uniendo en *clusters* por niveles.

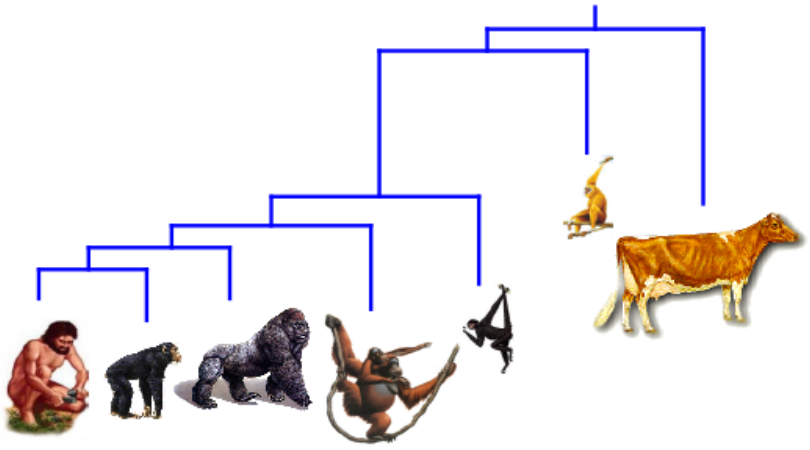

Figura 3.1: Ejemplo de dendograma.

El dendograma nos puede servir para decidir cuál es el valor de k y nos da una idea de si existen *outliers*. Sin embargo, normalmente no será tan fácil.

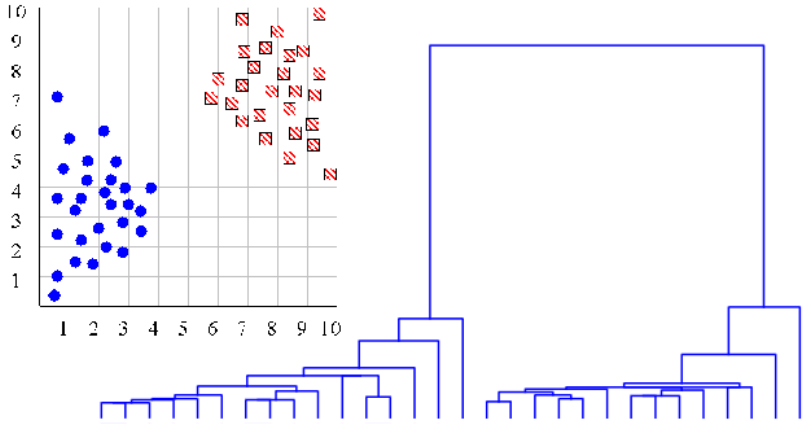

Figura 3.2: Dendograma indicando $k = 2$.

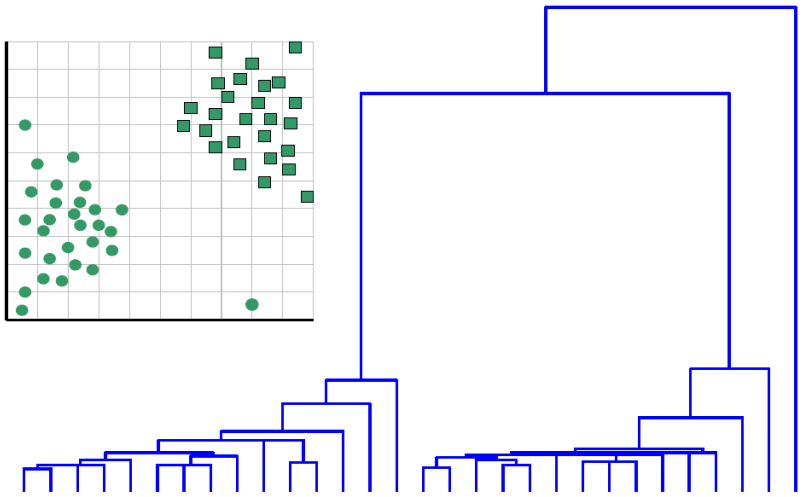

Figura 3.3: Dendograma indicando la existencia de un *outlier*.

3.2. Tipos de *clustering* jerárquicos

Existen dos tipos de *clustering* jerárquicos:

- **Aglomerativo:** la situación de partida suele ser un *cluster* por cada patrón. En cada paso se fusionan los dos *clusters* más cercanos. El proceso se repite hasta llegar a un único *cluster*.
- **Divisivo:** es el proceso contrario, se comienza con único *cluster* que se va dividiendo en cada paso hasta tener un *cluster* por patrón. Este proceso necesita de un mayor número de cálculos por lo que, por norma general, se aplica el aglomerativo.

Como se observa, la salida es una jerarquía entre *clusters*, donde el nivel de corte nos llevará a *clustering* distintos. De este modo, no es necesario fijar el valor de k.

3.3. Medir la distancia entre *clusters*

Una de las características a tener en cuenta en este tipo de *clustering* es determinar cómo se va a medir la distancia entre *clusters*. Aunque existen distintos métodos para enlazar *clusters*, nos quedaremos con los siguientes:

Enlace simple *(Simple Linkage):* la distancia entre dos *clusters* es la mínima distancia entre todos los posibles pares de objetos en ambos *clusters*.

$$D(C, C') = \min_{\boldsymbol{x} \in C, \boldsymbol{x}' \in C'} d(\boldsymbol{x}, \boldsymbol{x}')$$

Enlace completo *(Complete Linkage):* la distancia entre dos *clusters* es la máxima distancia entre todos los posibles pares de objetos en ambos *clusters*.

$$D(C, C') = \max_{\boldsymbol{x} \in C, \boldsymbol{x}' \in C'} d(\boldsymbol{x}, \boldsymbol{x}')$$

Enlace medio: es la distancia media entre todos los posibles pares de objetos dados dos *clusters*.

$$D(C, C') = \frac{1}{|C||C'|} \sum_{\boldsymbol{x} \in C, \boldsymbol{x}' \in C'} \boldsymbol{d}(\boldsymbol{x}, \boldsymbol{x}')$$

Centroide: la distancia entre dos *clusters* es la distancia entre sus centroides.

$$D(C, C') = \boldsymbol{d}(\boldsymbol{\mu_C}, \boldsymbol{\mu_{C'}})$$

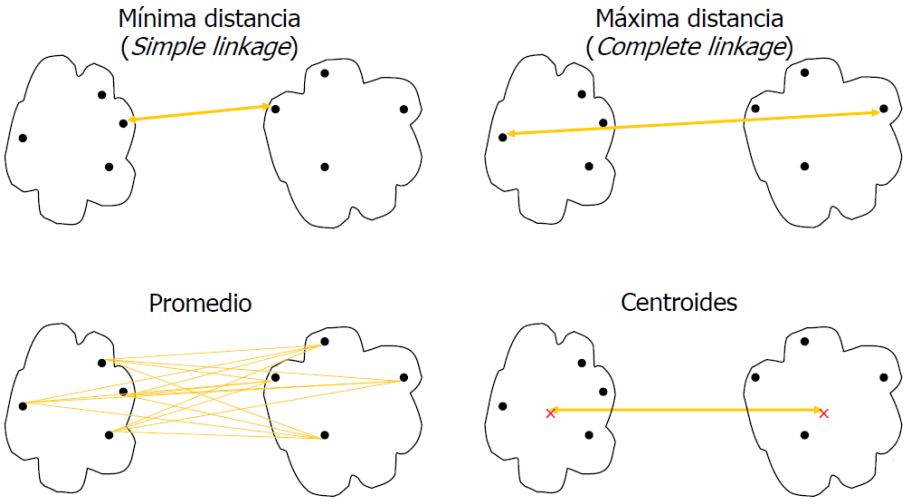

Figura 3.4: Tipos de distancia entre *clusters*.

3.4. Ventajas e inconvenientes

La principal desventaja de este tipo de *clustering* es que tiene una baja escalabilidad, es decir, para base de datos muy grandes el algoritmo es muy costoso. Aun así, presenta las siguientes ventajas:

- No es necesario establecer el valor de k, ya que se puede obtener mediante el dendograma.
- Permite la detección de *outliers*.
- Es determinista, es decir, no depende de la inicialización.
- Permite detectar *clusters* no convexos.

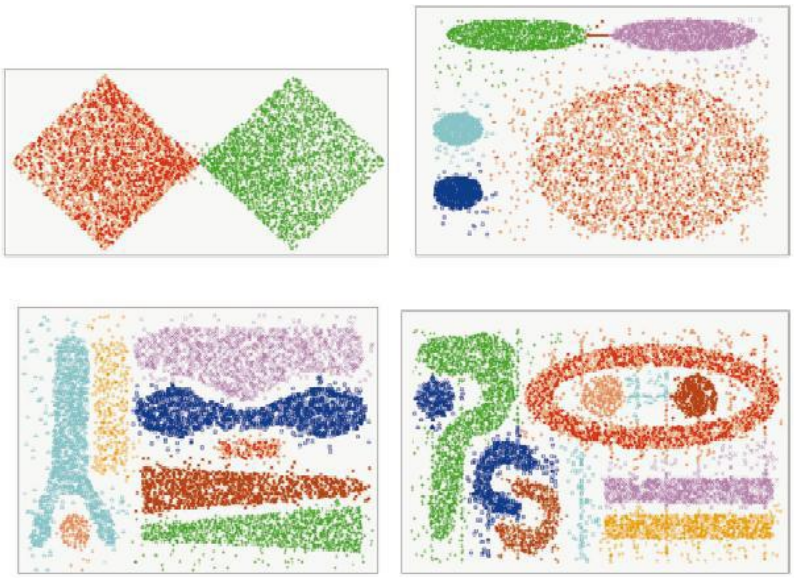

Figura 3.5: Detección *clusters* no convexos.

3.5. Ejemplo de aplicación

Dada la siguiente matriz de distancias entre cuatro patrones:

	A	**B**	**C**	**D**
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Se pide:

1. Aplicar un *clustering* jerárquico con el método de enlace simple.
2. Aplicar un *clustering* jerárquico con el método de enlace completo.

Solución

1. Enlace simple

Iteración 1

Elegimos la mínima distancia entre los *clusters:*

$d(A,B) = 1$ → es la mínima por lo que unimos esos dos puntos en el *cluster*. Tendríamos los siguientes *clusters*: $C1(A,B)$ – $C2(C)$ – $C3(D)$

Iteración 2

$d(C1,C2) = \min\big(d(A,C), d(B,C)\big) = \min(4,2) = 2$ → Unimos porque es la mínima.

$d(C1,C3) = \min\big(d(A,D), d(B,D)\big) = \min(5,6) = 5$

$d(C2,C3) = 3$

Tendríamos los siguientes *clusters*: $C1(A,B,C)$ – $C2(D)$

Iteración 3

Unimos los dos *clusters* que quedan: $C(A,B,C,D)$

Dendograma

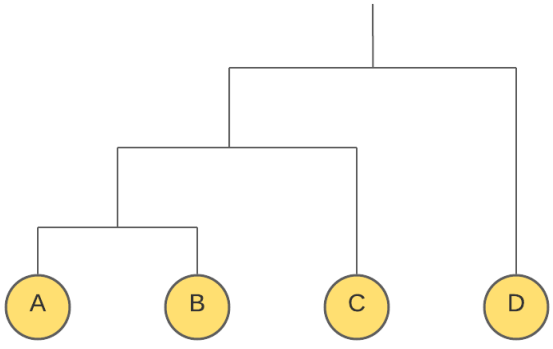

Figura 3.6: Dendograma usando enlace simple.

2. Enlace completo

Iteración 1

Elegimos la mínima distancia entre los *clusters:*

$d(A,B) = 1$ → es la mínima por lo que unimos esos dos puntos en el *cluster*. Tendríamos los siguientes *clusters*: $C1(A,B)$ – $C2(C)$ – $C3(D)$

Iteración 2

$d(C1,C2) = \max\big(d(A,C), d(B,C)\big) = \max(4,2) = 4$

$d(C1,C3) = \max\big(d(A,D), d(B,D)\big) = \max(5,6) = 6$

$d(C2,C3) = 3$ → Unimos porque es la mínima.

Tendríamos los siguientes *clusters*: $C1(A,B)$ – $C2(C,D)$

Iteración 3

Unimos los dos *clusters* que quedan: $C(A,B,C,D)$

Dendograma

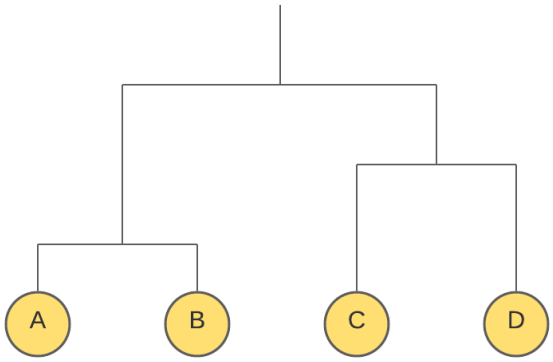

Figura 3.7: Dendograma usando enlace completo.

Vemos que la principal diferencia entre ambos se produce en la iteración dos. Esto es, cuando se van a establecer dos *clusters* para el agrupamiento. Con el simple tenemos $C1(A,B,C)$ – $C2(D)$, mientras que con el completo tenemos $C1(A,B)$ – $C2(C,D)$. Esto resulta, como se puede ver en los cálculos, por la forma de calcular la distancia entre los *clusters*.

4. AGRUPAMIENTO BASADO EN DENSIDADES

El agrupamiento basado en densidades se fundamenta en el siguiente principio: agrupar regiones densas de puntos separadas de otras regiones densas mediante regiones poco densas.

A diferencia del agrupamiento basado en densidades, los algoritmos anteriores no son capaces de obtener buenos resultados cuando se aplican a tareas con *clusters* de forma arbitraria o *clusters* dentro de *clusters.*

Es capaz de detectar *outliers*, es decir, regiones de baja densidad cuyos puntos no serán introducidos en ningún *cluster*.

4.1. Algoritmo DBSCAN

DBSCAN significa *Density-based Spatial Clustering of Applications with Noise*. El algoritmo trabaja con la idea de que, si un punto pertenece a un *cluster*, éste debería estar rodeado de un montón de otros puntos en ese *cluster*.

Tiene dos parámetros muy importantes. El radio ϵ determina la longitud específica para mirar alrededor del punto seleccionado. Por otro lado, el número de puntos mínimos M que debe haber alrededor de otra observación para definir un *cluster*.

Para cada punto se establece de que tipo es de entre los tres siguientes:

- Dado un punto al azar, miramos si ese punto es un punto **central.** Es decir, dentro de la vecindad definida por el radio ϵ, hay al menos M puntos.
- Un punto es **fronterizo** si su vecindario contiene menos de M puntos y se puede acceder desde algún punto central. En otras palabras, que dentro de su vecindario haya un punto **central**.

- Un valor **atípico** es un punto que no es punto central, y tampoco está lo suficientemente cerca como para ser accesible desde un punto central.

Una vez que el algoritmo ha decidido de que tipo es cada punto, se conectan todos los puntos centrales que son vecinos, en un mismo *cluster*. Así, un *cluster* estará formado por un punto central, todos los puntos centrales accesibles y los puntos fronterizos.

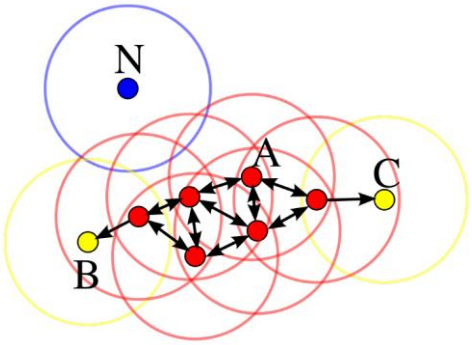

Figura 4.1: Puntos centrales (rojo), fronterizos (amarillo) y *outlier* (azul).

4.2. Ventajas e inconvenientes

De todo lo anterior se deduce que:

- Identifica *clusters* de forma arbitraria.
- Son robustos a la presencia de ruido.
- No es necesario fijar el valor k, pero sí que se tiene que fijar ϵ y M.
- Es escalable, ya que sólo necesita un único recorrido sobre el conjunto de datos.
- Un problema es que los puntos fronterizos pertenezcan a más de un *cluster* y, por tanto, se tenga que decidir a que *cluster* pertenecen resultando un algoritmo no determinista.
- Si existen grandes diferencias en las densidades, DBSCAN puede que no trabaje bien ya que es complicado escoger un conjunto de parámetros que funcionen bien para todos los grupos.

4.3. Ejemplo de aplicación

Dados los mismos puntos del ejemplo 2.3.

Se pide:

1. Aplicar DBSCAN considerando $M = 2$ y $\epsilon = \sqrt{2}$.
2. Aplicar DBSCAN considerando $M = 2$ y $\epsilon = \sqrt{10}$.

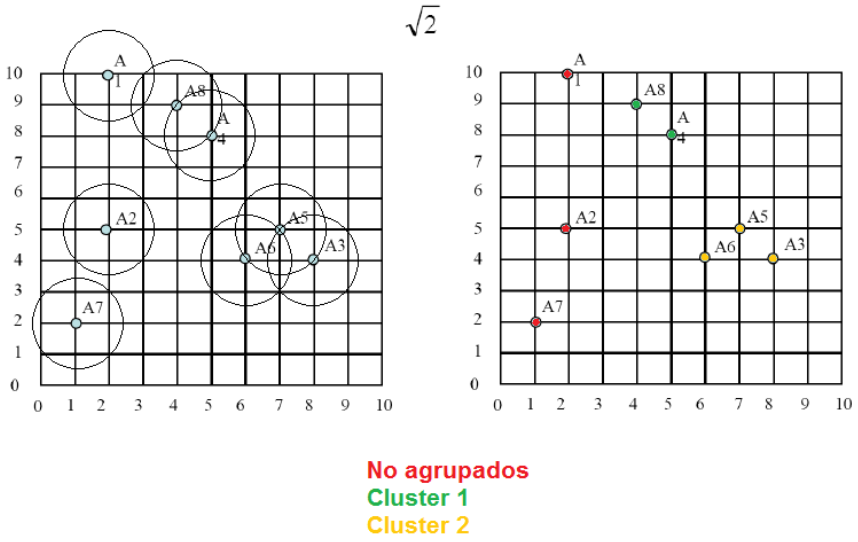

Figura 4.2: Resultado de aplicar DBSCAN considerando $M = 2$ y $\epsilon = \sqrt{2}$.

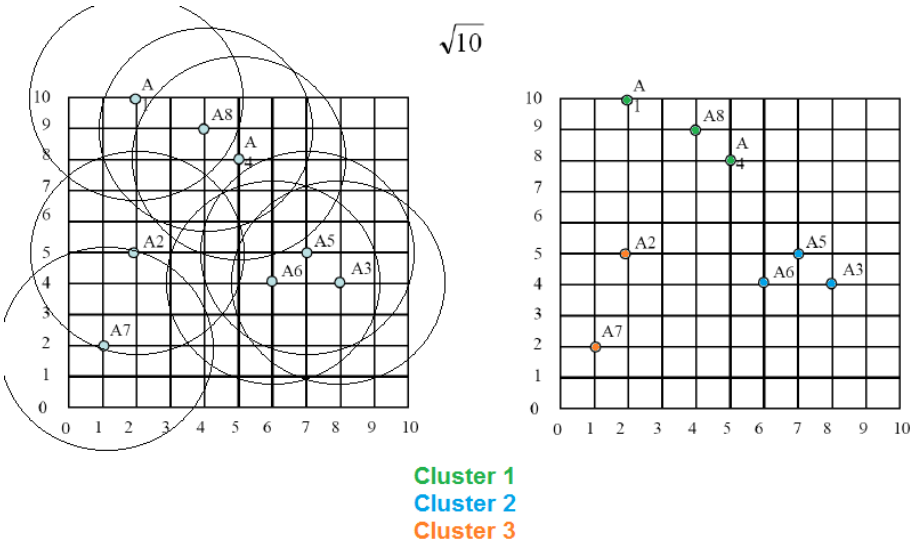

Figura 4.3: Resultado de aplicar DBSCAN considerando $M = 2$ y $\epsilon = \sqrt{10}$.

5. PUNTOS CLAVE

Una vez que hemos desarrollado los puntos más importantes del aprendizaje no supervisado, y más concretamente del *clustering*, estaremos capacitados para:

- Conocer la motivación del aprendizaje no supervisado.
- Aplicar los distintos algoritmos de *clustering* entendiendo sus parámetros.
- Conocer sus ventajas e inconvenientes.
- Interpretar los resultados obtenidos.